Psychtoolbox 3.0

Routines for Psychophysics
Outline

• Hardware and Software
• PTB installation
• Testing and Timing
• Basic functions
• Examples and Exercises
• Miscellaneous
Hardware and Software

• Requirements
• Displays
  LCD
  CRT
  Projectors
• Matlab, GNU/Octave
Hardware and Software

Basic functionality available on any graphics hardware:

Basic OpenGL 1.0 support required.

16 MB VRAM required.

High speed and advanced features:

OpenGL 1.5, better 2.0/2.1 capable.

> 128 MB VRAM, more is better.

Simplifies a lot of complex stimulus programming tasks. Allows for high precision, e.g., 32 bpc floating point.

PTB-3 regularly tested on: NVidia GeForce, ATI Radeon. Onboard chips are very limited in functionality and speed! (e.g. Intel MacBooks, iMacs)
Old Graphics Hardware

- Graphics card == Mostly passive image store.
- Psychtoolbox draws directly into Framebuffer while buffer is scanned out for display.
- Most drawing operations implemented/executed in software on slow CPU.
- Only very few operations are hardware-accelerated (Image copy, filled rectangles).
Modern Graphics Hardware

- GPU's are massive parallel and highly programmable.
- High memory bandwidth: > 100 GB/s vs. 8 GB/s on CPU.
- Computational speed: 475 GFlops vs. 12-24 GFlops.
- PTB interfaces with the GPU mostly via OpenGL.

Here used to be a picture of the tremendously complex ATI Radeon HD 2900 GPU, as an example of how complex a modern graphics processor is.

However, we don't have permission (yet) to publish this picture...
PTB-3: Double Buffered Drawing

1) Subject sees cloudy picture on the display screen (==Frontbuffer).
2) Matlab issues Screen() drawing commands.
3) Graphics hardware draws into backbuffer, processing the OpenGL commands in the background while Matlab and Psychtoolbox are able to do other stuff in parallel, e.g., keyboard queries, sound output...
The time course of the phosphor decay is an essential
time to control stimulus durations.

The time course of the phosphor decay is an essential
time to control stimulus durations.
Figure 2. Comparison of CRT and LCD luminance transition signals.

The plots show the onset of a green stimulus on a black monitor at time $t_0$. The CRT signal shown in (a) was measured from an Iiyama HM 204 DT monitor. The LCD signal shown in (b) was taken from a Dell 3007 WFP panel after backlight filtering by the division method [13]. Note the different scalings of the abscissas.

doi:10.1371/journal.pone.0012792.g002

Figure 3. Response times variability over different luminance levels (Dell 3007 WFP LCD panel, green channel).

The bar plots show averaged response times over five measurements. Response times have been measured from five different initial levels (given as percentage of the maximal luminance of the monitor) to five target levels. All response times have been calculated by means of the division method with dynamical filtering [13].

doi:10.1371/journal.pone.0012792.g003
presentation. We assume the classical disk–ring paradigm which was introduced already decades ago [24]. Variations of this paradigm were used in several of the studies from the literature review, both for CRT monitors [25] and for LCD monitors [26].

In our hypothetical experiment, a disk stimulus is presented for one frame. In the next frame, a surrounding ring is presented, and our hypothetical experimenter specifies both the stimulus duration and the interstimulus interval (ISI) between disk and ring according to the SOF method. As for the ISI, the SOF method assumes temporal adjacency, that is the onset of the ring occurs at the same time as the offset of the disk. The experimenter would therefore state an ISI of zero and a duration of the disk of one frame, as outlined in the top row of Fig. 6. If we assume the refresh rate which was most frequently used in the reviewed studies, namely 60 Hz (see Fig. 5A), the experimenter would specify a stimulus duration of 16.7 ms.

Let us compare these assumptions with the signal shapes on a CRT monitor, as sketched in the bottom row of Fig. 6. The luminance signal of the disk stimulus is a pulse determined by the almost instantaneous phosphor activation at frame start and by the following phosphor decay to zero. Obviously, it is unclear how to specify the true stimulus duration. One might integrate over the pulse and approximate a rectangular signal of the equivalent luminance or one might define the offset at the point in time when the luminance has decayed to 10% of its maximum, or define a time $p$ from the beginning of the frame until the luminance has decayed to nearly zero, as Bridgeman [9] suggested. In Fig. 6 we do the latter. In our sketch, $p = 7$ ms, which is in the time range of typical CRT phosphors [6]. The true disk duration would be $2p = 14$ ms now instead of 16.7 ms, that is, the SOF method would considerably overestimate the duration. Note that there are monitors with much lower phosphor decay times (see, for instance, Fig. 2(a)). In addition, the assumption of temporal adjacency is
PTB Installation

- for Windows
- for Mac OS/X
- for Linux
Basic Functions

• OpenWindow()
• Screen()
• try - catch
• Priority()
• AssertOpenGL
• VBLSyncTest
Screen() 

Controls all aspects of the graphics- and display-hardware.

• Performs all 2D drawing operations.
• Controls stimulus onset timing and provides timestamps.
• Allows for some high performance image processing.
• Performs on-demand stimulus post-processing.
try
% main prg

catch
% catch error: This is executed in case something goes wrong
% in the 'try' part due to programming error:

% Do same cleanup as at the end of a regular session...
Screen('CloseAll');
ShowCursor;
fclose('all');
Priority(0);

% Output the error message that describes the error:
psychrethrow(psychlasterror);

end
screens = Screen('Screens');
screenNumber = max(screens);

gray = GrayIndex(screenNumber);

[w, wRect] = Screen('OpenWindow', screenNumber, gray);
Screen(w, 'BlendFunction', GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);

Screen('Flip', w);

tex = Screen('MakeTexture', w, imdata);

Screen('DrawTexture', w, tex);

[VBLTimestamp start] = Screen('Flip', w);

Screen('CloseAll');
Priority()

% Returns as default the mean gray value of screen:
gray=GrayIndex(screenNumber);

% Open a double buffered fullscreen window on the screen
% 'screenNumber' and choose a gray background. 'w' is the handle
% used to direct all drawing commands to that window - "Name" of
% the window. 'wRect' is a rectangle defining the size.
% See "help PsychRects"
[w, wRect]=Screen('OpenWindow',screenNumber, gray);

% Set priority for script execution to realtime priority:
priorityLevel=MaxPriority(w);
Priority(priorityLevel);
% Make sure keyboard mapping is the same on all operating systems
% Apple MacOS/X, MS-Windows and GNU/Linux:
KbName('UnifyKeyNames');

% assign response key
resp=KbName('b');  % respond with key 'b'

% initialize KbCheck and variables to make sure they're
% properly initialized/allocted by Matlab - this to avoid time
% delays in the critical reaction time measurement part of the
% script:
[KeyIsDown, endrt, KeyCode]=KbCheck;

while  ( KeyCode(resp)==0)
    [KeyIsDown, endrt, KeyCode]=KbCheck;
    WaitSecs(0.001);
end
Example and Exercises
Example and Exercises

- Display image HolidaySnap.jpg
- HolidaySnap1.m
Example and Exercises

• Display image HolidaySnap.jpg
  HolidaySnap1.m

• Draw an anti-aliased fixation dot
  HolidaySnap2.m
Example and Exercises

• Display image HolidaySnap.jpg
  HolidaySnap1.m

• Draw an anti-aliased fixation dot
  HolidaySnap2.m

• Record response time and response
  HolidaySnap3.m
Micellaneous

• Textures and Alpha Blending
  AlphaImageDemo.m

• QuickTime movie playback
  SimpleMovieDemo.m

• Image processing pipeline
  ProceduralGaboriumDemo.m

• 3D Graphics OpenGL
  UtahTeapotDemo.m

• PsychPortAudio
Summary

Three Commandments of Experimentation:

• Know thy stimuli
• Know thy methods
• Know thy observers